(2) Japanese Patent Application Laid-Open No. JP8-115206(1996)

"Floating Point Number Operation Device"

The following is an extract relevant to the present application.

A floating point number operation device including a means for adding three selecting circuits, three shifters, a complement circuit, and two data buses and providing an add/subtract circuit with an output of a multiply/divide normalization circuit by the first two bits thereof, a means for complementing an adder/subtracter adder circuit input data, and a means for bit shifting the adder/subtracter adder circuit input data, an adder/subtracter adder circuit result and an adder circuit overflow to be returned to the adder/subtracter adder circuit input again.

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号
特開平8－115206
(43)公開日　平成８年(1996)５月７日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　7/552	Ｂ			
7/00				
7/52	３１０　Ｃ			
		8323－5Ｅ	Ｇ０６Ｆ　7/00　１０１　Ｗ	

審査請求　未請求　請求項の数１　ＯＬ　（全 14 頁）

(21)出願番号　特願平6－249089
(22)出願日　平成６年(1994)10月14日

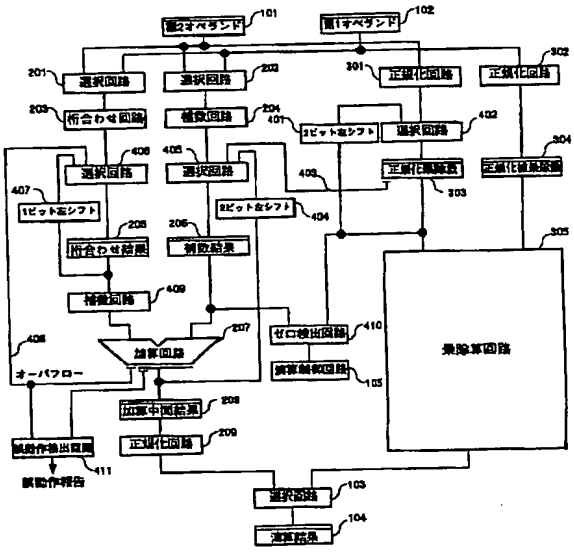
(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(71)出願人　000233011
日立コンピュータエンジニアリング株式会社
神奈川県秦野市堀山下１番地
(72)発明者　瀧口　誠
神奈川県秦野市堀山下１番地　日立コンピュータエンジニアリング株式会社内
(74)代理人　弁理士　小川　勝男

最終頁に続く

(54)【発明の名称】　浮動小数点演算装置

(57)【要約】
【目的】　一般的な構成の浮動小数点演算装置に少量の回路を付加することにより平方根演算器が可能となる浮動小数点演算装置を提供する。
【構成】　３個の選択回路、３個のシフタ、補数回路、２本のデータバスを付加し、乗除算器正規化回路の出力を先頭から２ビットづつ加減算回路に供給する手段、加減算器加算回路入力データを補数化する手段、加減算器加算回路入力データと加減算器加算回路結果と加算回路オーバフローをビットシフトし再び加減算器加算回路入力に戻す手段を設けた浮動小数点演算装置。
【効果】　一般的な構成の浮動小数点演算装置に少量の回路を付加することにより、非回復型平方根アルゴリズムによる平方根演算可能となる。

図１



1

【特許請求の範囲】

【請求項１】　加減算器と乗除算器とで構成される浮動小数点演算装置において、前記乗除算器の正規化乗除数レジスタの先頭２ビットを前記加減算器に転送する第１データバスと、前記加減算器に転送した２ビットのデータをシフトアウトして正規化乗除数レジスタにセットするための第１の２ビット左シフタ及び選択回路と、前記加減算器の加算回路の出力結果を２ビット左シフトする第２の２ビット左シフタと、加減算器の補数回路の出力、第２の２ビット左シフタの出力、及び乗除算器からの正規化乗除数レジスタの先頭２ビットをマージしたデータを選択する加減算器の補数結果レジスタ入力選択回路と、加減算器の加算回路の結果の最上位ビットを加減算器の桁合わせ結果レジスタ入力選択回路に転送する第２データバスと、加減算器の桁合わせ結果レジスタを１ビット左シフトする第３の１ビット左シフタと、加減算器の桁合わせ回路の出力、第３の１ビット左シフタの出力、及び加算回路の結果の最上位ビットをマージしたデータを選択する加減算器の桁合わせ結果レジスタ入力選択回路と、加減算器の桁合わせ結果レジスタの出力を補数化する桁合わせ結果のレジスタ出力補数回路と、加減算器の補数結果レジスタと乗除算器の正規化乗除数レジスタがゼロであるか否かを検出し、演算制御回路にゼロ情報を送出するゼロ検出回路とを備えたことを特徴とした浮動小数点演算装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、浮動小数点演算装置に関し、特に、既存の演算器回路を最大限に使用し、最小限の回路を付加することにより平方根演算を可能にする浮動小数点演算装置に関するものである。

【０００２】

【従来の技術】従来、平方根の計算をハードウエアで実現する場合の手法として非回復型平方根アルゴリズムが知られている。これは、図９に示すように、加減算とシフトにより１回の開平ステップで１ビットの部分平方根を求める手法である。この手法による平方根演算については、例えば「コンピュータの高速演算方法」（堀越彌訳、近代科学社、昭和５５年９月１日発行）第３５７頁から３５９頁に論じられている。

【０００３】非回復型平方根アルゴリズムと冗長２進表現による平方根専用演算器を用い、１回の開平ステップで２ビットの部分平方根を求める発明が特開平３－２９６１３２公報に開示されている。

【０００４】非回復型平方根アルゴリズムと類似している手法に回復型平方根アルゴリズムがある。これは、図１０に示すように、減算とシフトにより平方根を求める手法である。この手法の平方根演算に関するものには、例えば特開平１－１５５４３５公報が挙げられる。特開平１－１５５４３５公報では回復型平方根アルゴリズム

2

を除算器上で実現した平方根演算器の発明について開示されている。これは回復型平方根アルゴリズムと高基数非回復型除算アルゴリズムの手法が類似しているため除算器に必要な回路を付加して平方根演算を実行可能にしたものである。前記図９及び図１０において、２は加減算器、３は乗除算器、１０１は第２オペランドレジスタ、１０２は第１オペランドレジスタ、１０３，２０１，２０２は選択回路、１０４は演算結果レジスタ、２０３は桁合わせ回路、２０４は補数回路、２０５は桁合わせ結果レジスタ、２０６は補数結果レジスタ、２０７は加算回路、２０８は加算中間結果レジスタ、２０９，３０１，３０２は正規化回路、３０３は正規化乗除数レジスタ、３０４は正規化被乗除数レジスタ、３０５は乗除算回路である。

【０００５】

【発明が解決しようとする課題】従来技術の非回復型平方根アルゴリズムと冗長２進表現による発明は、１回の開平ステップで２ビットの部分平方根を求めることが可能であるため性能的には優れているが専用演算器を設けるためにハードウエア物量が増加してしまう。

【０００６】科学技術計算の分野で用いられる平方根演算命令は、命令処理の高速化も要求されるが、命令の使用頻度が比較的少ないため専用演算器を設けることによる性能向上の効果よりハードウエア物量増加の悪影響の方が大きい場合がある。

【０００７】従来技術の除算器上で平方根演算を実行する発明は、既存の演算器に平方根演算で必要な回路を付加するためハードウエア物量の増加は少量ですむが、高基数非回復型除算アルゴリズムと異なる除算アルゴリズムの除算器では実現不可能な場合がある。例えば、乗算器を用いた収束型除算器では実現不可能である。

【０００８】本発明の目的は、一般的な構成の浮動小数点演算装置に少量の回路を付加することにより、平方根演算が可能となる浮動小数点演算装置を提供することにある。

【０００９】本発明の前記ならびにその他の目的及び新規な特徴は、本明細書の記載及び添付図面によって明らかになるであろう。

【００１０】

【課題を解決するための手段】本願において開示された発明のうち代表的なものの概要を簡単に説明すれば、以下のとおりである。

【００１１】加減算器と乗除算器とで構成される浮動小数点演算装置において、前記乗除算器の正規化乗除数レジスタの先頭２ビットを前記加減算器に転送する第１データバスと、前記加減算器に転送した２ビットのデータをシフトアウトして正規化乗除数レジスタにセットするための第１の２ビット左シフタ及び選択回路と、前記加減算器の加算回路の出力結果を２ビット左シフトする第２の２ビット左シフタと、加減算器の補数回路の出力、

第2の2ビット左シフタの出力、及び乗除算器からの正規化乗除数レジスタの先頭2ビットをマージしたデータを選択する加減算器の補数結果レジスタ入力選択回路と、加減算器の加算回路の結果の最上位ビットを加減算器の桁合わせ結果レジスタ入力選択回路に転送する第2データバスと、加減算器の桁合わせ結果レジスタを1ビット左シフトする第3の1ビット左シフタと、加減算器の桁合わせ回路の出力、第3の1ビット左シフタの出力、及び加算回路の結果の最上位ビットをマージしたデータを選択する加減算器の桁合わせ結果レジスタ入力選択回路と、加減算器の桁合わせ結果レジスタの出力を補数化する桁合わせ結果のレジスタ出力補数回路と、加減算器の補数結果レジスタと乗除算器の正規化乗除数レジスタがゼロであるか否かを検出し、演算制御回路にゼロ情報を送出するゼロ検出回路とを備えたものである。

【0012】また、信頼性を確保した遅延時間の短縮は、前記の加減算器の加算回路の結果最上位ビットのかわりに、加減算器の加算回路の結果のオーバフローを加減算器の桁合わせ結果レジスタ入力選択回路の転送データバスとして用いること、及び加減算器の加算回路の結果最上位ビットの値と加算回路の結果オーバフローの値を比較する誤動作検出回路を設けたものである。

【0013】

【作用】前述の手段によれば、乗除算器の正規化乗除数レジスタの先頭2ビットを加減算器に転送し、この加減算器に転送した2ビットのデータを2ビット左シフタと選択回路でシフトアウトして正規化乗除数レジスタにセットし、前記加減算器の加算回路の結果を2ビット左シフトし、加減算器の補数回路の出力、加減算器の加算回路の結果の2ビット左シフタ出力、乗除算器からの正規化乗除数レジスタの先頭2ビットをマージしたデータを選択し、加減算器の加算回路の結果の最上位ビットを加減算器の桁合わせ結果レジスタ入力選択回路に転送し、加減算器の桁合わせ結果レジスタを1ビット左シフトし、加減算器の桁合わせ結果レジスタ入力選択回路で加減算器の桁合わせ回路出力、加減算器の桁合わせ結果レジスタの1ビット左シフタは出力、及び加算回路の結果の最上位ビットをマージした（重ねた）データを選択し、桁合わせ結果レジスタ出力補数回路で加減算器の桁合わせ結果レジスタ出力を補数化し、ゼロ検出回路で加減算器の補数結果レジスタと乗除算器の正規化乗除数レジスタがゼロであるか否かを検出し、演算制御回路にゼロ情報を送出することにより、加減算器と乗除算器から構成される一般的な浮動小数点演算装置に、3個のシフタ、補数回路、2個のデータバスという少量の回路を付加するだけで、平方根演算可能な浮動小数点演算装置を実現することができる。

【0014】また、さらに加算回路の結果の最上位ビットとオーバフロー値を比較する誤動作検出回路を設けることにより加算回路のオーバフローの誤動作を検出することができる。

【0015】

【実施例】以下、本発明による実施例を図面を用いて詳細に説明する。

【0016】図1は、本発明による一実施例の浮動小数点演算装置の仮数部の概略構成を示すブロック図である。図1において、101は第2オペランドレジスタ、102は第1オペランドレジスタ、103は加減算器と乗除算器の演算結果を選択する選択回路、104は演算結果を格納する演算結果レジスタ、105は演算器全体の制御を行なう演算制御回路である。

【0017】201～209は、ロード命令や加減算命令を実行する加減算器の構成要素であり、201は指数小側の仮数データを選択する選択回路、202は指数大側の仮数データを選択する選択回路、203は指数小側仮数データの桁合わせを行なうシフタ、204は指数大側仮数データを真の減算時に補数化する補数回路、205は桁合わせ回路出力を保持する桁合わせ結果レジスタ、206は補数回路出力を保持する補数結果レジスタ、207は桁合わせデータと補数回路データを加算する加算回路、208は加算回路の結果を保持する加算中間結果レジスタ、209は加算中間結果データを正規化する正規化回路である。

【0018】301～305は、乗除算命令を実行する乗除算器の構成要素であり、301は第2オペランドを正規化する正規化回路、302は第1オペランドを正規化する正規化回路、303は正規化回路301の出力を保持する正規化乗除数レジスタ、304は正規化回路302の出力を保持する正規化被乗除数レジスタ、305は正規化乗除数と正規化被乗除数から乗除算結果を求める乗除算回路である。401～411は、本発明により浮動小数点演算装置に付加した平方根演算のための平方根演算回路であり、401は正規化乗除数レジスタの値を2ビット左シフトする第1の2ビット左シフタ、402は正規化回路301出力、第1の2ビット左シフタ401出力を選択する正規化乗除数レジスタ入力選択回路である。

【0019】403は正規化乗除数レジスタ303に格納されたa_{2k+1}とa_{2k+2}を選択回路405に転送するデータバスである。404は加算回路207の出力R_kを2ビット左シフトする第2の2ビット左シフタである。405は補数回路204出力、データバス403、第2の2ビット左シフタ404の出力を選択する補数結果レジスタ入力選択回路である。

【0020】406は加算回路207のオーバフロービットを選択回路408に転送するデータバスである。407は桁合わせ結果レジスタ205に格納したq_1～q_{k-1}を1ビット左シフトする第3の1ビット左シフタである。408は桁合わせ回路203出力、加算回路207のオーバフローデータバス406、第3の1ビット左シ

フタ４０７の出力を選択する桁合わせ結果レジスタ入力選択回路である。

【００２１】４０９は桁合わせ結果レジスタ２０５の出力をｑｋの値により補数化する桁合わせ結果レジスタ補数回路である。４１０は補数結果レジスタ２０６、正規化乗除数レジスタ３０３に格納されているＲｋとａ2k+1～ａnがゼロであることを検出するゼロ検出回路である。ゼロ検出回路４１０で検出されたゼロ情報は、演算制御回路１０５に転送され繰返し演算の終了条件に使われる。

【００２２】４１１は平方根演算の繰返し演算時に加算回路２０７の結果の最上位ビットの値とオーバフローの値を比較する誤動作検出回路である。加算回路２０７の結果の最上位ビットとオーバフロー値が排他的でないとき誤動作報告を行なう。

【００２３】一般的な浮動小数点演算装置の加算器は、仮数部７バイトにガードデジット０.５バイトを合わせた７.５バイト（（１，０）～（８，３））の加減算を可能とするため７.５バイト幅のデータを処理できる。

【００２４】図２は、本実施例の前記正規化乗除数レジスタ入力選択回路４０２の動作を説明するための図である。この正規化乗除数レジスタ入力選択回路４０２は、図２に示すように、平方根命令以外（乗除算命令）と平方根命令の初期設定時は正規化回路の出力を選択し、平方根演算の繰返し演算時は、２ビット左シフタ４０１の値を選択する。２ビット左シフタ４０１と選択回路４０２により平方根演算時、正規化乗除数レジスタ３０３にはａ2k+1～ａnの値を左詰めで格納できる。そのため選択回路４０５にはａ2k+1とａ2k+2、ゼロ検出回路４１０にはａ2k+3～ａnを容易に供給できる。

【００２５】図３は、本実施例の前記補数結果レジスタ入力選択回路４０５の動作を説明するための図である。この補数結果レジスタ入力選択回路４０５は、図３に示すように、平方根命令以外（加減算、ロード命令）時は正規化回路の出力を選択し、平方根演算の初期設定時はデータバス４０３のａ1、ａ2を選択する。平方根演算の繰返し演算時には、上位ビットに２ビット左シフタ４０４出力、下位ビットにはデータバス４０３のａ2k+1、ａ2k+2を選択する。平方根演算でｋ＞ｎ＋１が成立した場合の終了処理では丸め処理を行なうために、ｑ57と同じビット位置に１を立てるため０……０１００を選択する。平方根演算のＲｋ＝０、かつａ2k+1～ａn＝０が成立した途中終了の場合の終了処理時はゼロを選択する。

【００２６】図４は、前記桁合わせ結果レジスタ入力選択回路４０８の動作を説明するための図である。この桁合わせ結果レジスタ入力選択回路４０８は、図４に示すように、平方根命令以外（加減算、ロード命令）時は桁合わせ回路の出力選択し、平方根演算の初期設定時はゼロを選択する。平方根演算の繰返し演算時、終了処理時は上位ビットに１ビット左シフタ４０７の出力のｑ1～ｑk-1を選択し、次のビットにはオーバフローバス４０６のｑｋを選択し、最下位の２ビットには０を選択する。

【００２７】図５は、前記桁合わせ結果レジスタ補数回路４０９の動作を説明するための図である。この桁合わせ結果レジスタ補数回路は平方根命令以外（加減算、ロード命令）時、平方根演算の終了時は桁合わせ結果レジスタ出力を選択する。平方根演算の繰返し演算でｑｋ＝１時は桁合わせ結果レジスタの出力をビットごとに反転し、最下位の２ビットを１としたデータを選択する。平方根演算の繰返し演算でｑｋ＝０時は桁合わせ結果レジスタの出力の最下位の２ビットを１としたデータを選択する。

【００２８】図６は、本実施例の平方根演算器で実行する平方根命令の処理手順を示すフローチャートであり、長精度浮動小数点平方根命令の処理フローであるが、短精度浮動小数点平方根命令の場合には被開平数と演算結果の仮数が５６ビットから２４ビットに変わるだけである。

【００２９】本実施例で演算する平方根命令は、図６に示すように、最初に、被開平数が正、ゼロ、負であるか調べる（ステップ６０１）。負の場合には平方根例外の割込みを報告する（ステップ６０６，６０７）。ゼロの場合には演算処理を行なわず演算結果をゼロにして終了する（ステップ６０８）。正の場合には被開平数を指数が偶数になる様に正規化する（ステップ６０２）。

【００３０】これは次に行なう指数の演算処理（ステップ６０３）、すなわち、正規化した指数の値を２で割り演算結果指数を求める処理（ステップ６１１）において割り切れるよう、あらかじめ指数を偶数とする必要があることによる。仮数の演算処理（ステップ６０４）は仮数平方根を求める処理（ステップ６１２）と丸め処理（ステップ６１３）から構成される。

【００３１】仮数平方根を求める処理（ステップ６１２）は非回復型平方根アルゴリズムで５７ビットの平方根を求めることにより実現される。５７ビットの平方根を求めるのは、その後に行なう丸め処理（ステップ６１３）のためで、演算結果より１ビット余分に平方根を求める。

【００３２】丸め処理（ステップ６１３）では、余分に求めた最終ビットに１を加える。最後に、指数演算処理と仮数演算処理の演算結果を結合した値が演算結果となり（ステップ６０５）、平方根命令演算を終了する。

【００３３】図７は、本実施例の浮動小数点演算装置の動作を模式的に示した図である。図７では、選択回路で選択したデータがわかるように記述したため選択回路自身の記述を省略した。

【００３４】平方根演算では、桁合わせ結果レジスタ２０５は、ｑ1～ｑk００を格納するために使用し、補数結果レジスタ２０６はＲｋを格納するために使用する。ま

た、補数回路４０９の出力が（±Ｄｋ）となる。

【００３５】初期設定動作を説明する。図７に示すように、最初に、第２オペランドレジスタ１０１に被開平数がセットされる。次に、正規化回路３０１で指数が偶数になるように、被開平数仮数部を正規化し、そのデータを正規化乗除数レジスタ３０３にセットする。次に、桁合わせ結果レジスタ２０５には０をセットし、補数結果レジスタ２０６には上位ビットを０とし最下位の２ビットには正規化乗除数レジスタ３０３の最上位の２ビット（ａ１、ａ２）をセットする。

【００３６】正規化乗除数レジスタ３０３は、ａ１、ａ２を加算器に転送すると、２ビット左シフタ４０２によりａ１、ａ２をシフトアウトしデータａ３～ａｎを格納する。正規化乗除数レジスタ３０３の値はＲ０となる。ここまでが初期設定動作である。

【００３７】次に、繰返し演算動作を説明する。桁合わせ結果レジスタ２０５にセットされているｑ１～ｑｋはｑｋの値に従い補数回路４０９で補数化し（±Ｄｋ）とする。補数回路４０９での補数化は、先に説明した図５に示すように行なう。補数回路４０９の出力（±Ｄｋ）は、補数結果レジスタ２０６にセットされているＲｋと加算回路２０７で加算する。

【００３８】加算回路２０７の結果はＲｋ+1であり、加算回路２０７のオーバフロー４０６はｑｋ+1である。加算回路２０７の結果Ｒｋ+1は、２ビット左シフタ４０４でシフトされ、正規化乗除数レジスタ３０３の最上位２ビットａ2k+1、ａ2k+2を最下位ビットに付加して補数結果レジスタ２０６にセットする。

【００３９】正規化乗除数レジスタ３０３は、ａ2k+1、ａ2k+2を加算器に転送すると２ビット左シフタ４０２により、ａ2k+1、ａ2k+2をシフトアウトしデータａ2k+3～ａｎを格納する。加算回路２０７のオーバフロー２０６であるｑｋ+1は、桁合わせ結果レジスタ２０５を１ビット左シフタ４０７でシフトしたｑ１～ｑｋ-1の下位ビットに付加して桁合わせレジスタ２０５にセットする。

【００４０】誤動作検出回路４１１は、加算回路２０７の結果の最上位ビットとオーバフロー値を比較し、これらが排他的でないとき誤動作報告を行なう。ここまでが繰返し演算の１回分である。繰返し演算は１マシンサイクルで１回演算し、繰返し演算１回では１ビットのｑｋが求まる。繰返し演算をｎ回繰返しｑ１からｑｎを求める。本実施例の浮動小数点演算装置で演算する長精度平方根命令では５７回繰返しｑ１からｑ57を求める。

【００４１】次に、終了処理動作を説明する。終了動作はｋ＞ｎ＋１（図７の非回復型平方根アルゴリズムではｋ＞ｎまでだが、本実施例の浮動小数点演算装置で演算する平方根命令では１ビット余分にｑｋを求め、丸め処理を行なうためｋ＞ｎ＋１とする）が成立した場合、及びＲｋ＝０、かつａ2k+1～ａｎ＝０が成立した途中終了の場合がある。

【００４２】ｋ＞ｎ＋１が成立した場合には、ｑｎ+1のビットに１を加える丸め処理を行なう。丸め処理では、最初に補数結果レジスタ２０６に０……０１００を選択してセットする。

【００４３】次に、補数回路４０９は、桁合わせレジスタに格納されているｑ１～ｑｎ+1を素通りさせ、その出力と補数回路２０６の０……０１００を加算回路２０７で加算する。２０７の加算結果は、加算中間結果レジスタ２０８にセットする。次に、加算中間結果レジスタ２０８にセットした値を正規化回路２０９でｑ１が最も左のビットに来るようにシフトし、ｑｎより右のビットを切捨て演算結果レジスタにセットし演算を終了する。

【００４４】Ｒｋ＝０、かつａ2k+1～ａｎ＝０が成立した途中終了の場合は、最初に、補数結果レジスタ２０６にゼロを選択してセットする。次に、補数回路４０９は桁合わせレジスタに格納されているｑ１～ｑｋを素通りさせ、その出力と補数回路２０６のゼロを加算回路２０７で加算する。加算回路２０７の加算結果は、加算中間結果レジスタ２０８にセットする。次に、加算中間結果レジスタ２０８にセットした値を正規化回路２０９でｑ１が最も左のビットに来るようにシフトし演算結果レジスタ１０４にセットし演算を終了する。

【００４５】以上の説明からわかるように、本実施例によれば、乗除算器の正規化乗除数レジスタの先頭２ビットを加減算器に転送し、この加減算器に転送した２ビットのデータを第１の２ビット左シフタ４０１及び選択回路４０２でシフトアウトして正規化乗除数レジスタ３０３にセットし、前記加減算器の加算回路２０７の結果を第２の２ビット左シフタ４０４で２ビット左シフトし、加減算器の補数回路２０６の出力、加減算器の加算回路２０７の結果を２ビット左シフトする第２の２ビット左シフタ４０４の出力、乗除算器の正規化乗除数レジスタ３０３の先頭２ビットをマージしたデータを選択し、加減算器の加算回路２０７の結果の最上位ビットを加減算器の桁合わせ結果レジスタ入力選択回路４０８に転送し、加減算器の桁合わせ結果レジスタ２０５を第３の１ビット左シフタ４０７で１ビット左シフトし、加減算器の桁合わせ結果レジスタ入力選択回路４０８で、加減算器の桁合わせ回路２０３の出力、第３の１ビット左シフタ４０７の出力、及び加算回路２０７の結果の最上位ビットをマージしたデータを選択し、桁合わせ結果レジスタ２０５の出力の補数回路４０９で加減算器の桁合わせ結果レジスタ２０５の出力を補数化し、ゼロ検出回路４１０で加減算器の補数結果レジスタ２０６と乗除算器の正規化乗除数レジスタ３０３がゼロであるか否かを検出し、演算制御回路１０５にゼロ情報を送出することにより、加減算器と乗除算器から構成される一般的な浮動小数点演算装置に、３個のシフタ、補数回路、２個のデータバスという少量の回路を付加するだけで、平方根演算可能な浮動小数点演算装置を実現することができる。

【００４６】また、さらに加算回路の結果の最上位ビットとオーバフロー値を比較する誤動作検出回路を設けることにより加算回路オーバフローの誤動作を検出することができる。

【００４７】図８は、本実施例の非回復型平方根アルゴリズム処理の手順を示すフローチャートである。

【００４８】図８おいて、ａ2k+1～ａnは、図１の正規化乗除数レジスタ３０３に左詰めで格納され、ａnより右のビットには０を格納する。ｑ1～ｑnは、図１の桁合わせ結果レジスタ２０５のビット（１，０）～（８，１）に右詰めで格納され、ｑ1より左のビットには０が入る。Ｒkは図１の補数結果レジスタ２０６に右詰めで格納され、上位のビットには符号を格納する。

【００４９】ステップ５２２のＲk+1＝Ｒk＋Ｄkまたは５１２のＲk+1＝Ｒk－Ｄkの計算は、図１の加算回路２０７でＲk+1＝Ｒk＋（±Ｄk）と変形しておこなう。（±Ｄk）は、図１の補数回路４０９へ桁合わせ結果レジスタ２０５に格納のｑ1～ｑkを入力し作成する。ｑk＝０の時（±Ｄk）は＋Ｄkとなり、その値は（ｑ1……ｑk１１）とする。ｑk＝１の時（±Ｄk）は－Ｄkとなり、その値は（ｑ1……ｑk０１）を２の補数化した値（￢ｑ1……￢ｑk１１）とする。

【００５０】平方根の繰返し演算時、加算回路２０７での計算が常に真の減算となることは、図８よりｑk＝０の時Ｒk+1＝Ｒk＋ＤkのＲkは負であり、ｑk＝１の時Ｒk+1＝Ｒk－ＤkのＲkは、正であることから明らかである。真の減算であるため加減算回路２０７の演算結果の最上位ビットがＲk+1の符号を、オーバフロー（４０６）は、Ｒk+1の￢符号を表す。Ｒk+1≧０の時ｑk+1＝１、Ｒk+1＜０の時ｑk+1＝０となることによりオーバフロー４０６＝ｑk+1となる。平方根の繰返し演算時は加算回路２０７の結果の最上位ビットのかわりにオーバフロー（４０６）を使用しても問題ない。

【００５１】図８のステップ５０３と５０４によるＲk＝０、ａ2k+1～ａn＝０の判定は、図１のゼロ検出回路４１０で行なう。ゼロ検出回路４１０で検出されたゼロ検出信号は、演算制御回路１０５に報告される。演算制御回路１０５はゼロ検出信号により平方根の繰返し演算を中止し終了処理を行なう。

【００５２】図９は、一般的な浮動小数点演算装置の仮数部の概略構成を示すブロック図であり、２は加減算器、３は乗除算器、１０１は第２オペランドレジスタ、１０２は第１オペランドレジスタ、１０３，２０１，２０２は選択回路、１０４は演算結果レジスタ、２０３は桁合わせ回路、２０４は補数回路、２０５は桁合わせ結果レジスタ、２０６は補数結果レジスタ、２０７は加算回路、２０８は加算中間結果レジスタ、２０９，３０１，３０２は正規化回路、３０３は正規化乗除数レジスタ、３０４は正規化被乗除数レジスタ、３０５は乗除算回路である。

【００５３】図９に示すように、通常の加算回路の演算結果は、パリティにより誤動作の検出を行なっているが、オーバフローなどのキャリーは、誤動作の検出を行なっていない。平方根演算の繰返し演算時は加算回路結果の最上位ビット＝加算回路のオーバフローが必ず成立することより、誤動作検出回路４１１は、加算回路の結果の最上位ビット＝加算回路オーバフローの時、誤動作報告を行なう。

【００５４】また、通常の加算回路は、その結果よりもオーバフローなどのキャリを早く求めるよう回路を設計する。そのため、加算回路２０７の結果の最上位ビットの符号を桁合わせ結果レジスタ入力選択回路に転送するかわりに、オーバフロー４０６を桁合わせ結果レジスタ入力選択回路に転送した方が遅延時間を短くできる。以上、本発明を実施例に基づき具体的に説明したが、本発明は、前記実施例に限定されるものではなく、本発明の要旨を逸脱しない範囲において、種々変更し得ることは勿論である。

【００５５】

【発明の効果】本願において開示された発明の代表的なものによって得られる効果を簡単に説明すれば、以下のとおれである。

【００５６】加減算器と乗除算器から構成される一般的な浮動小数点演算装置に３個のシフタ、補数回路、２個のデータバスという少量の回路を付加することにより、平方根演算可能な浮動小数点演算装置を実現することができる。

【００５７】また、加算回路の結果の最上位ビットとオーバフロー値を比較する誤動作検出回路を設けることにより、加算回路オーバフローの誤動作を検出することができる。

【図面の簡単な説明】

【図１】本発明による一実施例の浮動小数点演算装置の仮数部の概略構成を示すブロック図である。

【図２】本実施例の正規化乗除数レジスタ入力選択回路の動作を説明するための図である。

【図３】本実施例の補数結果レジスタ入力選択回路の動作を説明するための図である。

【図４】本実施例の桁合わせ結果レジスタ入力選択回路の動作を説明するための図である。

【図５】本実施例の桁合わせ結果レジスタ出力補数回路の動作を説明するための図である。

【図６】本実施例の浮動小数点演算装置で実行する浮動小数点平方根命令の処理手順を示すフローチャートである。

【図７】本実施例の浮動小数点演算装置の仮数部の動作を摸式的に示した図である。

【図８】本実施例の非回復型平方根アルゴリズムの処理手順を示すフローチャートである。

【図９】一般的な浮動小数点演算装置の概略構成を示す

11

ブロック図である。

【図１０】従来の回復型平方根アルゴリズムの処理手順を示すフローチャートである。

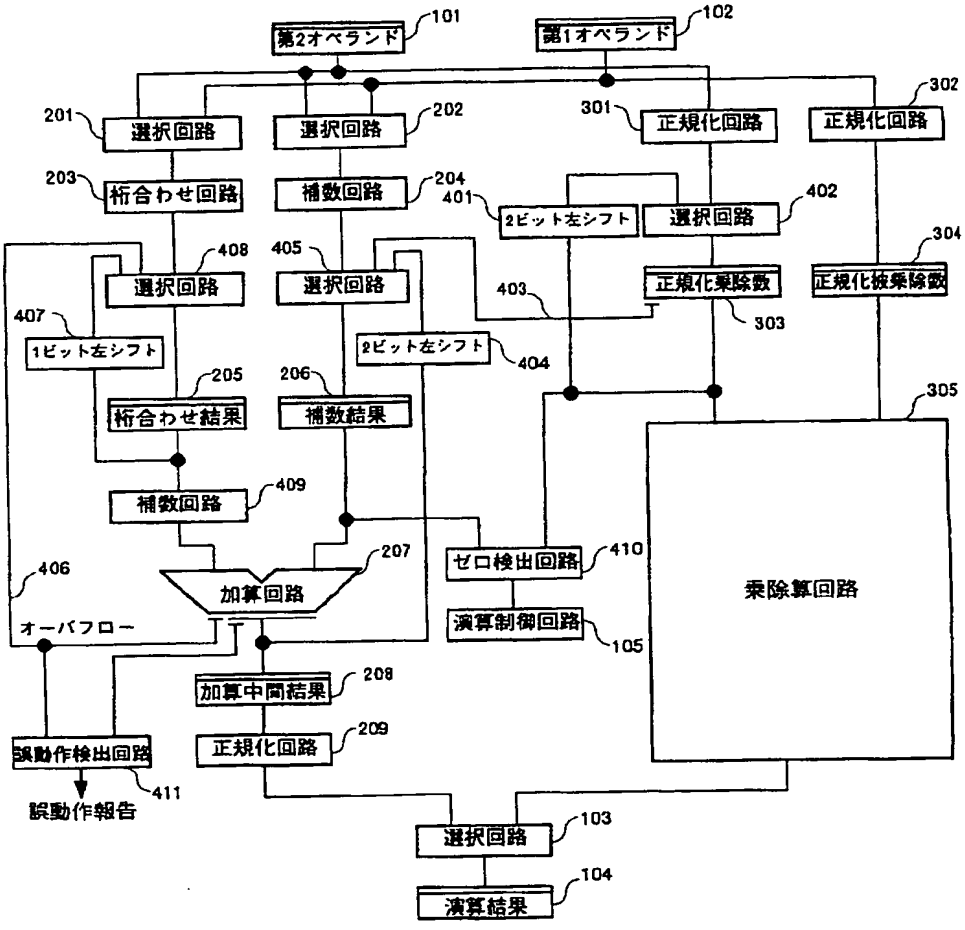
【符号の説明】

２…加減算器、３…乗除算器、１０１…第２オペランドレジスタ、１０２…第１オペランドレジスタ、１０３，２０１，２０２，４０２，４０５，４０８…選択回路、１０４…演算結果レジスタ、１０５…演算制御回路、２０３…桁合わせ回路、２０４，４０９…補数回路、２０＊５…桁合わせ結果レジスタ、２０６…補数結果レジスタ、２０７…加算回路、２０８…加算中間結果レジスタ、２０９，３０１，３０２…正規化回路、３０３…正規化乗除数レジスタ、３０４…正規化被乗除数レジスタ、３０５…乗除算回路、４０１，４０４…２ビット左シフタ、４０３，４０６…データバス、４０７…１ビット左シフタ、４１０…ゼロ検出回路、４１１…誤動作検出回路。

12

【図１】

図 1



【図２】

図２

バイト位置	1	2	3	4	5	6	7	8			
ビット位置	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0	1	2	3

選択条件	選択データ
平方根演算命令以外	正規化回路出力
平方根演算命令 初期設定時	正規化回路出力
繰り返し演算時	正規化乗除数レジスタ左2ビットシフト結果(a_{2K+1}～a_n) | 0 | 0

【図３】

図３

バイト位置	1	2	3	4	5	6	7	8			
ビット位置	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0	1	2	3

選択条件	選択データ
平方根演算命令以外	補数回路出力
平方根演算命令 初期設定時	ゼロ | a_1 | a_2
繰り返し演算時	加算回路出力左2ビットシフト結果(R_{k-1}) | a_{2k+1} | a_{2k+2}
丸め終了処理時	ゼロ | 1 | 0
途中終了処理時 (R_k=0 & a_{2k+1}～a_n=0)	ゼロ

【図４】

図４

バイト位置	1	2	3	4	5	6	7	8			
ビット位置	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0	1	2	3

選択条件	選択データ
平方根演算命令以外	桁合わせ回路出力
平方根演算命令 初期設定時	ゼロ
繰り返し演算時 終了処理時	桁合わせ結果レジスタ左1ビットシフト結果(q_1～q_{k-1}) | q_k | 0 | 0

【図５】

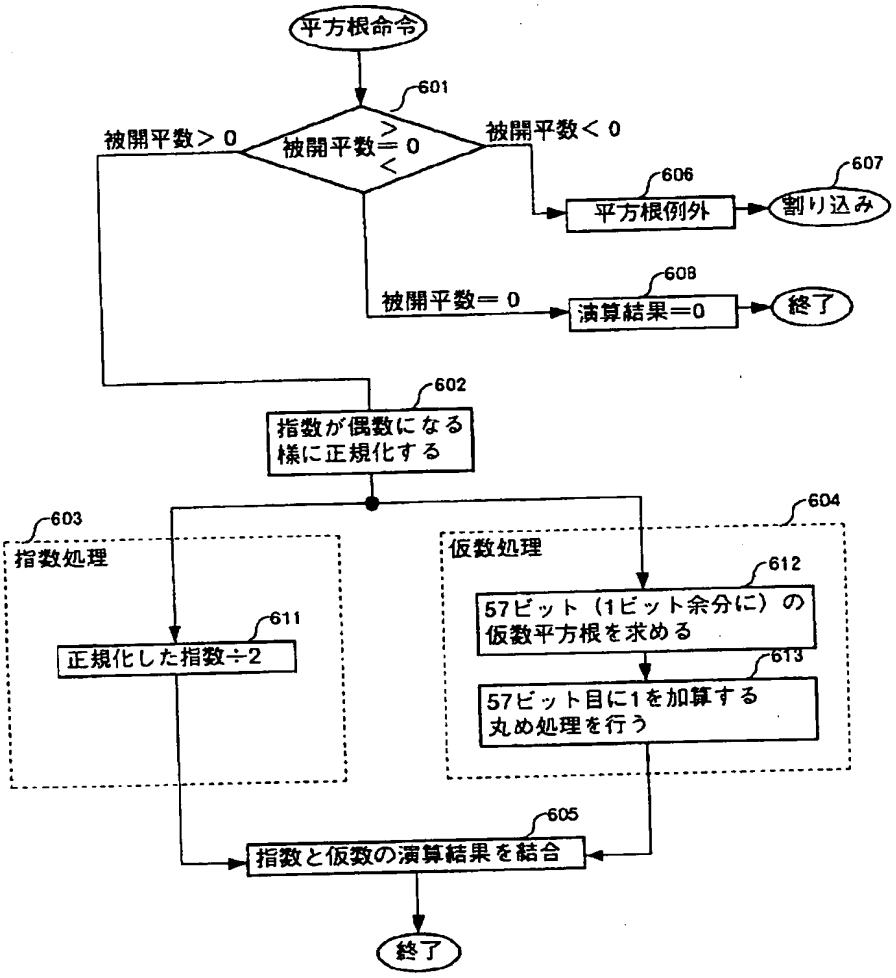
図５

バイト位置	1	2	3	4	5	6	7	8			
ビット位置	0-7	0-7	0-7	0-7	0-7	0-7	0-7	0	1	2	3

選択条件	選択データ
平方根演算命令以外	桁合わせ結果レジスタ出力
平方根演算命令 繰り返し演算q_K=1時	￣(桁合わせ結果レジスタ出力ビット(1,0)～(8,1) (q_1～q_K)) 1 1
繰り返し演算q_K=0時	(桁合わせ結果レジスタ出力ビット(1,0)～(8,1) (q_1～q_K)) 1 1
終了処理時	桁合わせ結果レジスタ出力 (q_1～q_K 0 0)

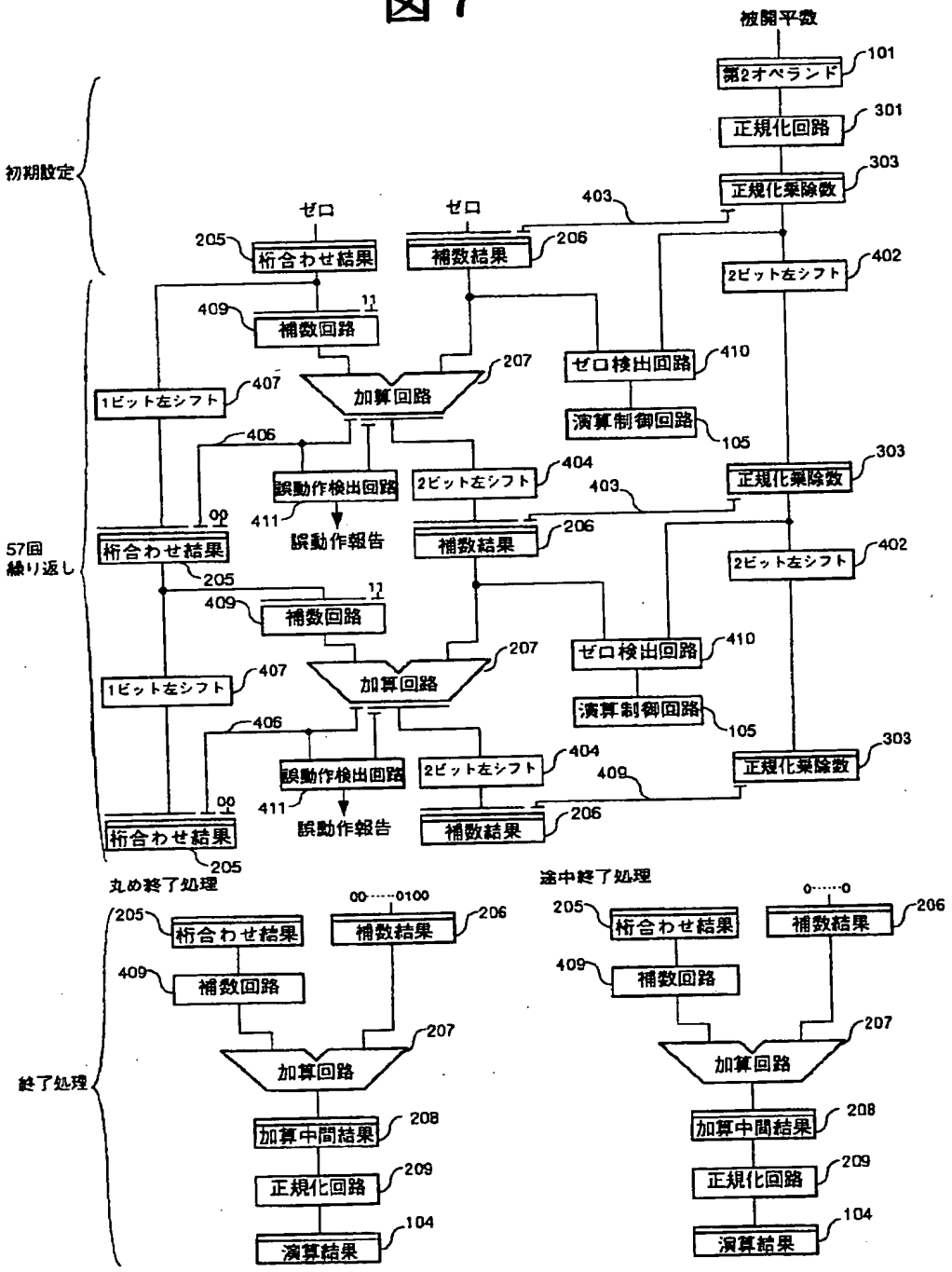
【図６】

図６


平方根命令
601
被開平数＞０
被開平数＝０
被開平数＜０
606
平方根例外
607
割り込み
被開平数＝０
608
演算結果＝0
終了
602
指数が偶数になる様に正規化する
603
指数処理
611
正規化した指数÷2
604
仮数処理
612
57ビット（1ビット余分に）の仮数平方根を求める
613
57ビット目に1を加算する丸め処理を行う
605
指数と仮数の演算結果を結合
終了

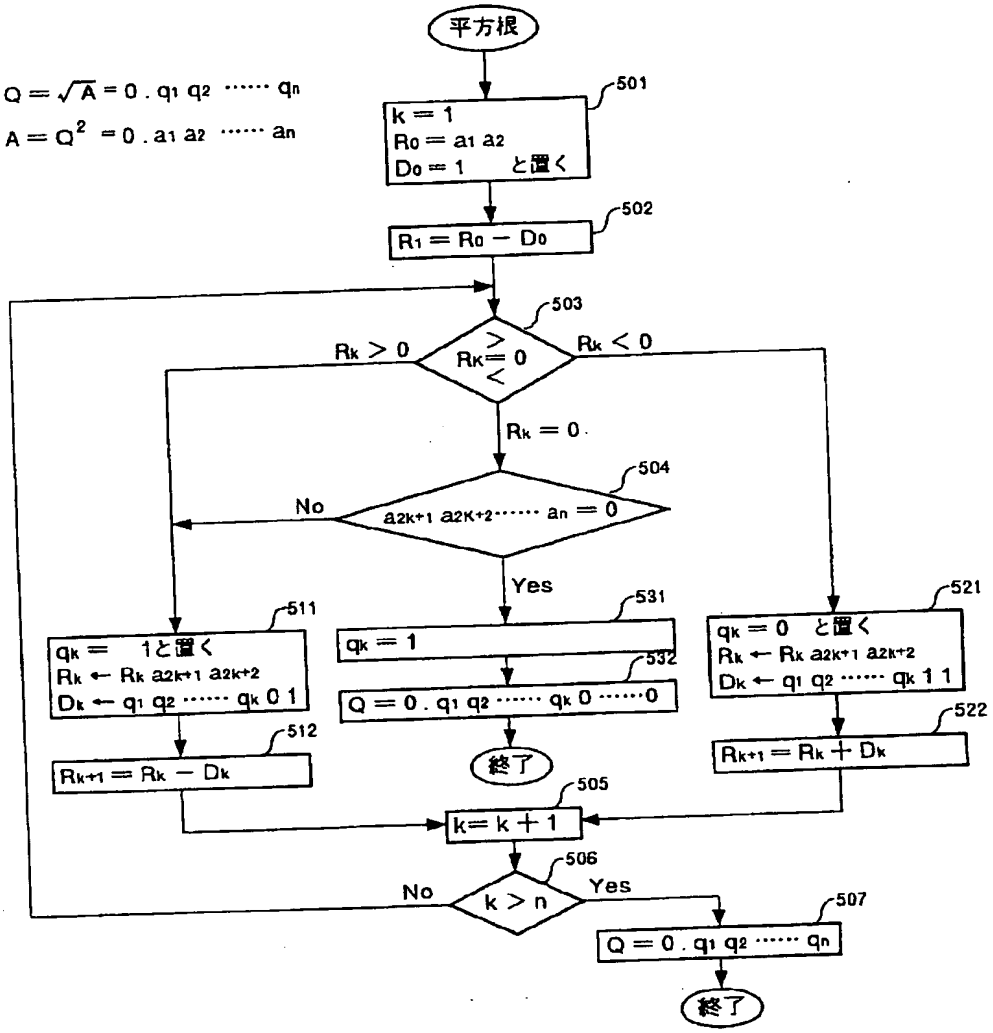
【図７】

図７


被開平数
第2オペランド
101
正規化回路
301
正規化乗除数
303
403
ゼロ
ゼロ
205
桁合わせ結果
補数結果
206
2ビット左シフト
402
409
補数回路
11
ゼロ検出回路
410
207
加算回路
1ビット左シフト
407
演算制御回路
105
406
誤動作検出回路
2ビット左シフト
404
411
誤動作報告
00
初期設定
57回
繰り返し
丸め終了処理
途中終了処理
00……0100
0……0
208
加算中間結果
209
正規化回路
104
演算結果
終了処理

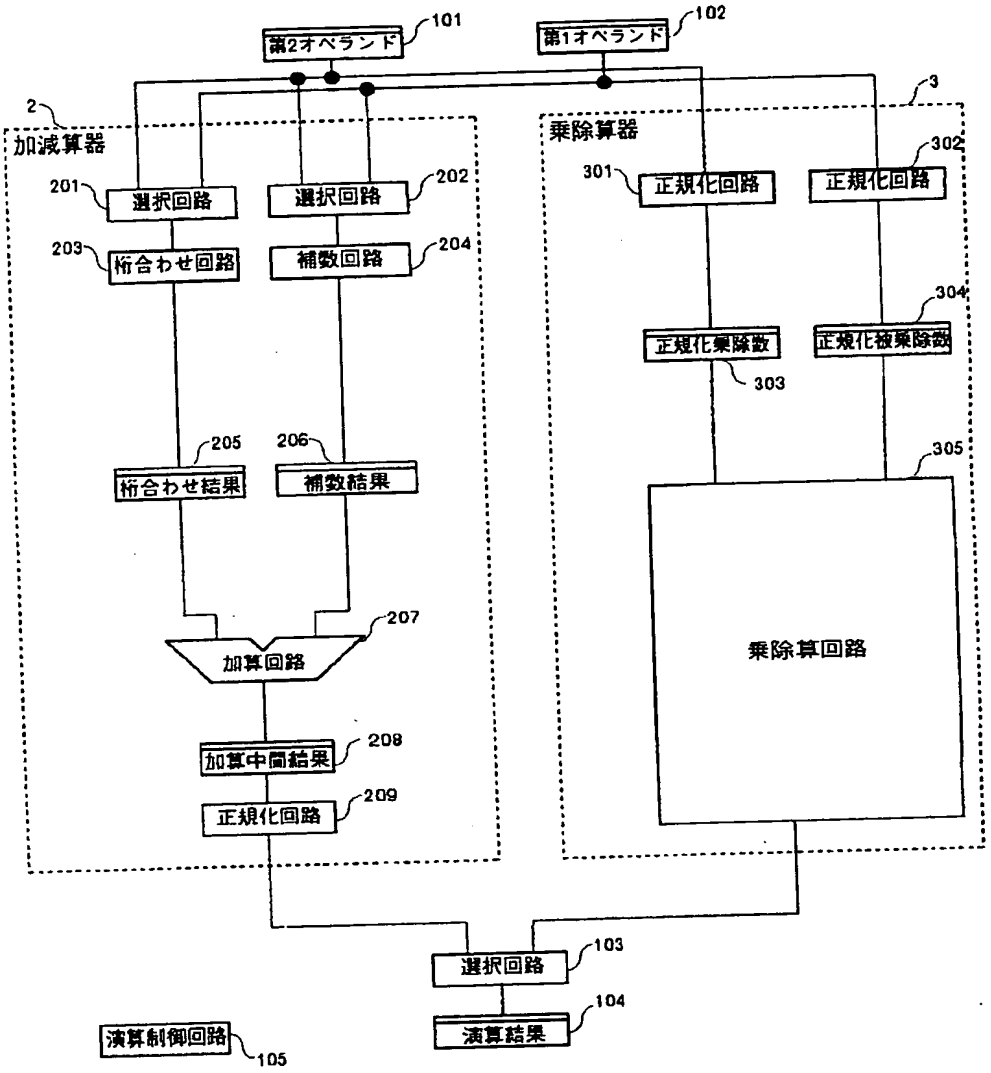
【図８】

図８


平方根
Q = √A = 0 . q1 q2 …… qn
A = Q2 = 0 . a1 a2 …… an
501
k = 1
R0 = a1 a2
D0 = 1　と置く
502
R1 = R0 − D0
503
Rk > 0
Rk = 0
Rk < 0
Rk = 0
504
No
a2k+1 a2K+2…… an = 0
Yes
511
qk = 1と置く
Rk ← Rk a2k+1 a2k+2
Dk ← q1 q2 …… qk 0 1
512
Rk+1 = Rk − Dk
531
qk = 1
532
Q = 0 . q1 q2 …… qk 0 ……0
終了
521
qk = 0　と置く
Rk ← Rk a2k+1 a2k+2
Dk ← q1 q2 …… qk 1 1
522
Rk+1 = Rk + Dk
505
k = k + 1
506
No
k > n
Yes
507
Q = 0 . q1 q2 …… qn
終了

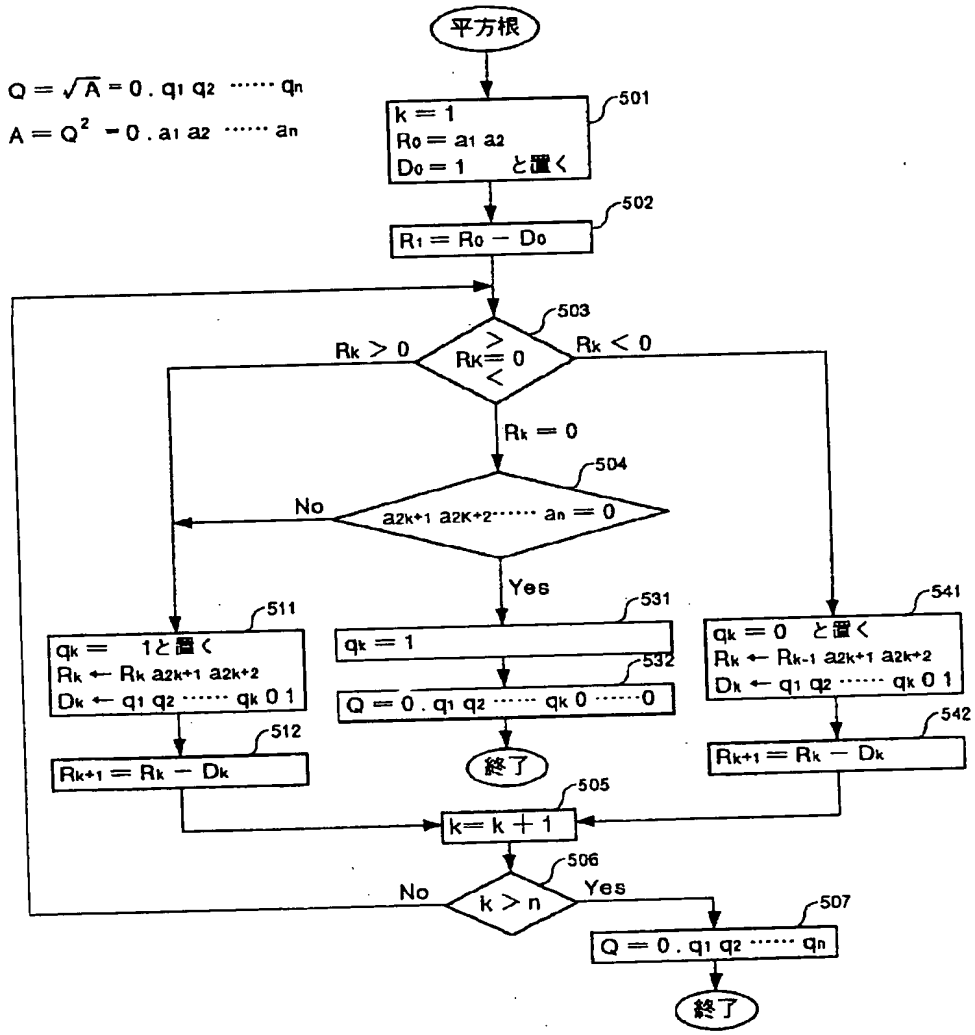
【図９】

図９


第2オペランド
101
第1オペランド
102
2
加減算器
3
乗除算器
201
選択回路
202
選択回路
203
桁合わせ回路
204
補数回路
301
正規化回路
302
正規化回路
304
正規化乗除数
正規化被乗除数
303
205
206
桁合わせ結果
補数結果
305
207
加算回路
乗除算回路
208
加算中間結果
209
正規化回路
103
選択回路
104
演算結果
演算制御回路
105

【図１０】

図１０



フロントページの続き

(72)発明者　増田　好徳
神奈川県秦野市堀山下１番地　日立コンピュータエンジニアリング株式会社内

(72)発明者　高橋　千秋
神奈川県秦野市堀山下１番地　日立コンピュータエンジニアリング株式会社内